1. (20 pts) Your company, Upstart Computers, has designed a non-pipelined version of the RISC-V processor. It currently uses the five stage datapath as described in the textbook (and therefore has a base CPI of 5). One of your employees has just found a way to optimize the hardware so that the clock speed can be doubled, and the MEM phase can be eliminated in those instructions that don’t need it. However, memory access speed has not changed, so at the new clock speed, all memory accesses now take twice as long in terms of clock cycles. The employee claims a two-fold increase in performance on a benchmark with the instruction mix shown below. Calculate the speedup of the new design and determine if the claim is correct.

<table>
<thead>
<tr>
<th>Instruction Type</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loads</td>
<td>20%</td>
</tr>
<tr>
<td>Stores</td>
<td>10%</td>
</tr>
<tr>
<td>ALU Ops</td>
<td>40%</td>
</tr>
<tr>
<td>Branches</td>
<td>20%</td>
</tr>
<tr>
<td>Other</td>
<td>10%</td>
</tr>
</tbody>
</table>

2. (15 pts) Hazard types. Below are several modifications that could be made to a processor implementation. Name the type(s) of pipeline hazard that the modification might influence, and state whether the effect would be to increase or decrease the likelihood of the hazard(s). If the modification would not effect a particular hazard type, say ‘NONE.’

- Increasing the clock speed by 20%.
- Increasing the depth (i.e., number of stages) of the execution pipeline.
- Fully pipelining an ALU that is currently not pipelined.
- Creating an “economy” version of the pipeline that reduces complexity by eliminating all data forwarding.
- Implementing a branch delay slot.
3. (5 pts) *Data forwarding* is a technique that can eliminate most, but not all, data hazards. For the five stage pipeline we studied in class, give an example where data forwarding *cannot* completely eliminate the hazard, and note where the hazard occurs.

4. (15 pts) When we first studied processor operation, we assumed that memory was always able to supply values during a single clock cycle. Since then, we have observed that this is probably only true if the underlying memory includes cache, and during cache misses a memory access will take longer than one cycle. Thus, the processor must be able to tolerate a memory system with a variable access time.

   - Explain how this complication affects a processor implemented using a five stage pipeline.
   - Explain how this complication affects a processor implemented using the Tomasulo algorithm.

5. (45 pts) Consider a processor that uses the “standard” five stage pipeline, except that floating point adds/subtracts take 2 execution cycles, and multiplies take 4 cycles; all integer instructions require one execute cycle. The processor uses a branch delay slot. Answer the following questions concerning the following code example:

   ```
   loop: FLD F0, 0(R1)
   FADD.D F4, F0, F2
   FLD F6, 0(R2)
   FMUL.D F2, F6, F4
   FSD F2, 0(R2)
   SUBI R2, R2, 8
   SUBI R1, R1, 8
   BNEQZ R1, loop
   NOP
   ```

   a) Show the execution of this code on a processor *without forwarding*. Show the pipeline timing diagram for one iteration of this loop, and determine the number of cycles it takes from the start of execution of the first load in the loop until the start of execution of the same load in the next iteration.

   b) For the processor implementation of part a), do not unroll the loop, but rearrange the instructions to minimize the number of cycles required to execute this loop. Calculate the speedup obtained from this rescheduling.

   c) Now repeat part a), except now modify the processor pipeline to assume all possible data forwarding paths. Calculate the speedup obtained from this change.